

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method ~~for accessing, in a computer system, a data object having an identifier (ID) and stored in a first storage location~~, comprising:
storing ~~the ID~~ an identifier (ID) in a second lock object, a data object having the ID being stored in a first storage location in a computer system;
determining whether the ID is contained in a first lock object;
if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and
if the link is assigned to the ID, ~~restricting~~ skipping performing a read and/or write access on the data object, and
if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and
if the ID is not contained in the first lock object, performing the read and/or write access on the data object; and
displaying to a user, on an output display, whether the read and/or write access on the data object is ~~restricted~~ skipped or can be performed.
2. (Previously Presented) The method of claim 1, wherein the first lock object is a file stored on a nonvolatile storage means.

3. (Previously Presented) The method of claim 1, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location.

4. (Previously Presented) The method of claim 1, wherein each data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

5. (Previously Presented) The method of claim 4, wherein the link is a filename or a link to a file.

6. (Previously Presented) The method of claim 1, wherein the first lock object is created by a data moving process.

7. (Previously Presented) The method of claim 1, wherein the second lock object is stored in a volatile storage means.

8. (Previously Presented) The method of claim 1, wherein the second lock object is a data array.

9. (Previously Presented) The method of claim 8, wherein the data array is one dimensional.

10. (Canceled).
11. (Currently Amended) A computer system for processing data, comprising:
memory means for storing program instructions;
input means for entering data;
storage means for storing data;
a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for accessing a data object having an identifier (ID) and stored in a first storage location, the method comprising:
storing the ID in a second lock object;
determining whether the ID is contained in a first lock object;
if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and
if the link is assigned to the ID, ~~restricting~~ skipping performing a read and/or write access on the data object, and
if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and
if the ID is not contained in the first lock object, performing the read and/or write access on the data object; and

displaying to a user, on an output display, whether the read and/or write access on the data object is ~~restricted~~ skipped or can be performed.

12. (Currently Amended) A computer readable storage medium comprising instructions for performing a method ~~for accessing a data object having an identifier (ID)~~ and ~~stored in a first storage location in a computer system~~, the method comprising:

storing ~~the ID~~ an identifier (ID) in a second lock object, a data object having the ID being stored in a first storage location in a computer system;

determining whether the ID is contained in a first lock object;

if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and

if the link is assigned to the ID, ~~restricting~~ skipping performing a read and/or write access on the data object, and

if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and

if the ID is not contained in the first lock object, performing the read and/or write access on the data object; and

displaying to a user, on an output display, whether the read and/or write access on the data object is ~~restricted~~ skipped or can be performed.

13-14. (Canceled).

15. (Currently Amended) The computer readable storage medium of claim 12, wherein the first lock object is a file stored on a nonvolatile storage means.

16. (Currently Amended) The computer readable storage medium of claim 12, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location.

17. (Currently Amended) The computer readable storage medium of claim 12, wherein each data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

18. (Currently Amended) The computer readable storage medium of claim 12, wherein the link is a filename or a link to a file.

19. (Currently Amended) The computer readable storage medium of claim 12, wherein the first lock object is created by a data moving process.

20. (Currently Amended) The computer readable storage medium of claim 12, wherein the second lock object is stored in a volatile storage means.

21. (Currently Amended) The computer readable storage medium of claim 12, wherein the second lock object is a data array.

22. (Currently Amended) The computer readable storage medium of claim 21, wherein the data array is one dimensional.

23. (Currently Amended) A computer system ~~for accessing a data object having an identifier (ID) and stored in a first storage location~~, comprising:

a processor responsive to program instructions;

means for storing ~~the ID~~ an identifier (ID) in a second lock object, a data object having the ID being stored in a first storage location; and

means for determining whether the ID is contained in a first lock object;

if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and

if the link is assigned to the ID, ~~restricting~~ skipping performing a read and/or write access on the data object, and

if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and

if the ID is not contained in the first lock object, performing the read and/or write access on the data object; and

~~means~~ an output display for displaying to a user, ~~on an output display~~, whether the read and/or write access on the data object is ~~restricted~~ skipped or can be performed.

24. (Previously Presented) The computer system of claim 23, wherein first lock object is a file stored on a nonvolatile storage means.

25. (Previously Presented) The computer system of claim 23, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location.

26. (Previously Presented) The computer system of claim 23, wherein each data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

27. (Previously Presented) The computer system of claim 23, wherein the link is a filename or a link to a file.

28. (Previously Presented) The computer system of claim 23, wherein the first lock object is created by a data moving process.

29. (Previously Presented) The computer system of claim 23, wherein the second lock object is stored in a volatile storage means.

30. (Previously Presented) The computer system of claim 23, wherein the second lock object is a data array.

31. (Previously Presented) The computer system of claim 30, wherein the data array is one dimensional.